

Started on	Thursday, 18 April 2024, 12:56 PM
State	Finished
Completed on	Thursday, 18 April 2024, 1:55 PM
Time taken	58 mins 44 secs
Marks	300.00/300.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Sebelumnya kalian sudah pernah mengerjakan kelas BunchOfKeys dalam bahasa pemrograman C++. Sekarang, kalian perlu mengimplementasikannya dalam bahasa Java. Berikut adalah header dari BunchOfKeys yang perlu ditranslasikan ke bahasa pemrograman Java

```
// BunchOfKeys.hpp
#ifndef __BUNCH_OF_KEYS_HPP__
#define __BUNCH_OF_KEYS_HPP__

#include iostream
using namespace std;

class BunchOfKeys {
public:
    // ctor
    BunchOfKeys();

    // member function
    void add();
    void shake();

private:
    int n_keys; // jumlah kunci yg ada
};

#endif
```

Catatan:

- ctor: menginit n_keys dengan nilai 0
- add(): menambah kunci (hanya melakukan increment pada atribut n_keys)
- shake(): mengeluarkan bunyi "kringing" sejumlah n_keys kali (setiap "kringing" diakhiri end-of-line) apabila jumlah kunci > 1. apabila jumlah kunci ≤ 1 , akan mengeluarkan pesan "Tidak terjadi apa-apa" (diakhiri end-of-line)
- Kumpulkan **BunchOfKeys.java**

Java 8

 [BunchOfKeys.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.06 sec, 28.82 MB
2	25	Accepted	0.07 sec, 28.55 MB
3	25	Accepted	0.06 sec, 28.82 MB
4	25	Accepted	0.07 sec, 28.89 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Diberikan tiga file, [MoneyInWallet.java](#), [MoneyInBank.java](#), dan [MoneyTransaction.java](#), yang mendefinisikan suatu interface transaksi keuangan, dan dua kelas yang mengimplementasikan interface tersebut.

Lengkapilah MoneyInWallet.java dan MoneyInBank.java sehingga memiliki implementasi method yang tepat.

Kumpulkan MoneyInWallet.java dan MoneyInBank.java dalam file **money.zip**

Java 8 ▾

 [money.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.35 sec, 28.00 MB
2	10	Accepted	0.22 sec, 27.87 MB
3	10	Accepted	0.32 sec, 29.01 MB
4	10	Accepted	0.26 sec, 28.98 MB
5	10	Accepted	0.19 sec, 28.20 MB
6	10	Accepted	0.27 sec, 27.86 MB
7	10	Accepted	0.15 sec, 27.84 MB
8	10	Accepted	0.14 sec, 28.44 MB
9	10	Accepted	0.14 sec, 28.21 MB
10	10	Accepted	0.14 sec, 29.10 MB

Question 3

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB



Terdapat interface [Calculable.java](#), static class [Comparator.java](#), dan abstract class [Hero.java](#). Kamu diminta untuk membuat 3 kelas baru yang merupakan turunan dari class Hero. Masing-masing kelas ini adalah Agility, Strength, dan Intelligence. Untuk mengimplementasikan kelas turunan tersebut, perlu dibuat file Java baru untuk setiap kelasnya

Kamu perlu mengimplementasikan turunan-turunan di atas untuk digunakan dalam kelas Comparator. Detail dari pengimplementasian kelas turunan di atas adalah sebagai berikut:

- Masing-masing kelas memiliki "nilai" yang lebih tinggi dari lain. Urutannya sendiri adalah Agility > Strength > Intelligence. Namun, hal ini hanya untuk ketika level dari masing-masing kelas sama. Ketika level berbeda, yang akan memiliki nilai tinggi mengikuti kepada level
- Method value() digunakan untuk menghitung nilai dari masing-masing object. Silahkan gunakan kreatifitas kamu untuk memenuhi constraint di atas
- Method printInfo() akan mengeluarkan "<TipeClass> Hero (Lvl <level>)" yang diakhiri oleh newline. Contoh untuk Agility hero yang memiliki level 15: **Agility Hero (Lvl 15)**
- Constructor perlu diimplementasikan

Berikut adalah contoh input dan output dari program

Input	Output	Keterangan
Hero1: Agility level 10 Hero2: Strength level 10 isObj1EqualToObj2(Hero1, Hero2)	false	Agility memiliki bobot yang lebih tinggi daripada Strength
Hero1: Intelligence level 20 Hero2: Agility level 10 isObj1BiggerThanObj2(Hero1, Hero2)	true	Hero1 memiliki level yang lebih tinggi
Hero1: Strength level 20 Hero1.printInfo()	Strength Hero (Lvl 20)	

* Kumpulkan file **Agility.java**, **Strength.java**, dan **Intelligence.java** dalam sebuah zip yang bernama **Hero.zip**

Java 8

[Hero.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.27 sec, 28.47 MB

No	Score	Verdict	Description
2	10	Accepted	0.24 sec, 28.18 MB
3	10	Accepted	0.13 sec, 26.20 MB
4	10	Accepted	0.14 sec, 28.40 MB
5	10	Accepted	0.11 sec, 28.38 MB
6	10	Accepted	0.11 sec, 29.88 MB
7	10	Accepted	0.11 sec, 28.66 MB
8	10	Accepted	0.14 sec, 28.10 MB
9	10	Accepted	0.09 sec, 28.30 MB
10	10	Accepted	0.23 sec, 29.01 MB

[◀ Slide Tutorial 4](#)

Jump to...



[Latihan Praktikum 4 ▶](#)